REMARKS

Applicants thanks Examiner for their detailed review of the application. Applicant has added a brief summary section, as requested by the Examiner.

Examiner objected to claim 31 due to informalities. Claim 31 has been amended to correct the antecedent basis problem changing the reference to "the first SMI handler" to "the SMI handler." Examiner also objected to use of "the second SMBase." However, in the first limitation of claim 31 applicant states, "changing a target SMBase of the SMI handler from the first SMBase address to **a second SMBase address** associated with the second processor." Consequently, applicant respectfully submits that use of "the second SMBase," in the second limitation of claim 31 is appropriate, as proper antecedent basis has been provided in claim 31.

Examiner rejects claims 1, 3-5, 45, and 47-49 under 102(a) as being anticipated by Dale (GB 2382180A). Dale discloses at page 18 lines 5-31 the following:

> The method is initiated by the expiration of one such timer in step 405, for example, when it is necessary to monitor the control channel as mentioned above. At the expiration of the timer, the GSM ULPD sets an RF_RADIO_ON signal (370 of FIG. 3) 'high', i.e. a voltage level '1', in step 410.
>
> ...
>
> The RF_RADIO_ON signal is also provided to the GPIO unit (321 of FIG. 3) of the MMI sub-system (320 of FIG. 3), generating an interrupt to the MMI processor (322 of FIG. 3), as illustrated in step 420. If necessary, the interrupt signal wakes the MMI processor. In step 425,

Here, Dale teaches providing a voltage signal to an RF regulator of a GSM sub-system and to a GPIO unit of a MMI sub-system. The GPIO unit of the MMI sub-system then generates an interrupt to wake MMI processor 322.

In contrast, applicant's claim 1 includes, "generating a wake-up signal **with the first processor after receiving the first SMI**," and "receiving the wake-up signal with the second

processor." Dale explicitly states an interrupt is generated by the GPIO unit **of the MMI subsystem** not a wake-up signal with the first processor as included in applicant's claim 1. Examiner equates Dale's Radio On signal to applicant's claim 1 usage of "first SMI;" however, Dale's GSM sub-system does not provide a wake-up signal after the Radio-on signal.

In addition, applicant's claim 1 includes, "the wake-up signal references a first memory address of a default SMI handler." Dale merely discloses recognizing the interrupt as originating from the GSM sub-system and initiating the wake up process, such as fetching and executing instructions from a memory element (See page 19 lines 1-5 and Figure 4 block 490), but does not teach or suggest either that the interrupt itself references a memory address of a default handler or that the instructions from memory element 150 are even associated with a handler. For at least the reasons stated above, applicant respectfully submits that claim 1 and its dependent claims 2-11 are now in condition for allowance.

Referring next to applicant's claim 45, the following limitations are included: "the first processor **executes the code at the first memory address**...and... after receiving the wake-up signal, wherein the second processor **executes the code at the first memory address**." Dale in Figure 4 illustrates that MMI processor is to initiate wake up of memory access components and GSM processor is to fetch and execute instruction from a memory element (See Figure 4). However, Dale does not suggest the MMI processor and the GSM processor are to execute code at the same memory location, i.e. code at the first memory address. In contrast, the first processor and the second processor in applicant's claim 45 execute code from the first memory address. As Dale does not suggest or teach executing code at the same location with a first and a second processor, applicant respectfully submits that claim 45 and its dependent claims, 46-53 are now in condition for allowance.

Examiner rejects claim 40 under U.S.C. 103 as being unpatentable over Dale in view of Nalawadi (US 2003.0009654 A1). However, claim 40 includes the limitation of "a first logical processor…to handle the first SMI and generate a wake-up signal **after the first SMI**. Examiner equates Dale's RADIO-ON signal to applicant's first SMI and interrupt from the GPIO as applicant's wake-up signal. However, as stated above, Dale expressly teaches that the GPIO is in the MMI sub-system. Therefore, Dale does not teach applicant's limitation that the first logical processor generates the wake-up signal, as Dales GSM sub-system only generates the RADIO-ON signal, not a wake-up signal that "references a first memory address of a default SMI handler." Also noted above, is that the GSM processor fetches and executes instructions from memory 350, but Dales does not disclose that the instructions are part of a default SMI handler and does not suggest that the location of those instructions is referenced in the RADIO-ON signal or interrupt generated by the GPIO. Additionally, Nalawadi is used in combination for logical processors; however, Nalawadi does not disclose the wake-up signal being generated by the first processor or that the wake-up signal references a default SMI handler. Consequently, applicant respectfully requests that independent claim 40 and its dependent claims, 41-44, are now in condition for allowance for the reasons stated above.

Next, Examiner rejects applicants 26-33, 35-38, 54-59, and 61 under 35 U.S.C 102(b) as being anticipated by Nguyen et al. (US 2002/0099893). Nguyen et al. discloses in paragraphs 0018 to 0019 the following:

> **82, 84**, and **86**, respectively. In one example, processor **12***b* is the processor that issued the software instruction that caused the issuance of the software SMI. The software SMI signature is saved to the SMRAM space **82** associated with processor **12***b*. In this example, processor **12***c* has been selected as the processor whose SMI handler will address the software SMI. Once processor **12***c* locates the software SMI signature **88**, which in this example is in the SMRAM space **82** associated with processor **12***b*, the SMI handler of processor **12***c* uses the parameters that have been passed to the SMRAM that includes software SMI signature. The parameter passing step **54** of **FIG. 2** is accomplished through the SMRAM space of the processor that caused the initiation of the software SMI, without regard to the processor that is selected by the BIOS of the computer system to handle the software SMI.
>
> [**0019**] In some cases, only one of the processors of a multiprocessor system is designated as being the processor that handles all system management interrupts. In this scenario, only one processor of the computer system, the SMI processor, will include a SMI handler and the responsibility for handling all system management interrupts will be passed to this processor. The disclosed method of parameter

Here, Nguyen discloses a method of passing parameters to a single processor executing an SMI handler. As noted, processor 12c is selected as the processor whose SMI handler will address the software SMI. Processor 12b is merely the processor that issued the initial instruction causing the generation of an SMI to be handled. Moreover, later Nguyen discloses that one processor may by dedicated as the processor to handle all SMIs, which entails only one processor executing SMI handler code.

In contrast applicant's claim 26 includes the limitation of, "executing a SMI handler with a first processor to handle the SMI for the first processor; and executing the SMI handler with a second processor to handle the SMI for the second processor." Unlike execution of SMI code on one processor with parameter passing from other processors, applicant's claims include execution of the SMI handler or SMI code on both the first and the second processor. Similarly, in claim 54, the limitations of "a first processor to execute the SMI code," and "a second processor to execute the

SMI code, if the SMI is software generated." Here, both the first and the second processor execute the SMI code if the SMI is software generated, instead of one processor handling a software SMI with parameter passing as described in Nguyen.

Furthermore, claim 32 includes a limitation of "checking if the SMI is a software generated SMI." However, Nguyen discloses at paragraph 0017:

> [**0017**] Shown in **FIG. 2** is a flow diagram of the process of handling a software SMI of the present disclosure. Prior to issuing a software SMI, at step **40**, the application program that is being run on one of the processors **12** performs a parameter passing function by writing the parameters of the SMI to one or more of the registers of the processor. At step **42**, the application program writes a unique software SMI signature to a register of the processor that is currently running the application program. At step **44**, the processor issues the software instruction that causes the initiation of the system management interrupt. In most instances, the software instruction executed by the processor is a write to a defined I/O port of the chip set, specifically a write to an I/O port of the PCI bridge **16** or the expansion bridge or south bridge **22**. At step **46**, the chip set receives the I/O write that was initiated by one of the processor and initiates the system management interrupt. In the system

As can be seen, Nguyen only discloses parameter passing before issuing a software SMI and does not disclose applicant's limitation of "checking if the SMI is a software generated SMI." Nguyen's step 46 in Figure 2 includes merely "the chip set receives the I/O write," not checking if the SMI is software generated. Additionally, in Nguyen an SMI signature is stored in SMRAM space and the SMI signature may be located by another processor (See paragraph 0018). However, this only describes Nguyen's method of parameter passing, i.e. passing an SMI signature parameter, to another processors SMRAM space, not checking whether a generated SMI is a software SMI, as compared to other SMIs, such as a hardware SMI.

Therefore, applicant respectfully submits that claims 26, 32, and 54, as well as their dependent claims 27-31, 32-39, and 55-61, are now in condition for allowance for the reasons stated above. Applicant again thanks Examiner for their detailed review, and notes that claims 12-25 were

previously designated as allowable. Therefore, applicant respectfully requests that claim 1-61 are in condition for allowance.

If there are any additional charges, please charge Deposit Account No. 50-0221. If a telephone interview would in any way expedite the prosecution of the present application, the Examiner is invited to contact David P. McAbee at (503) 712-4988.

Respectfully submitted,
Intel Corporation

Dated: January 11, 2006

/s/David P. McAbee/Reg. No. 58,104/
David P. McAbee
Reg. No. 58,104

Intel Corporation
M/S JF3-147
2111 NE 25th Avenue
Hillsboro, OR 97124
Tele – 503-712-4988
Fax – 503-264-1729